

(19) RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
PARIS

(11) N° de publication :
(à n'utiliser que pour les
commandes de reproduction)

2 752 122

(21) N° d'enregistrement national : 94 09357

(51) Int Cl⁶ : H 04 L 9/30

(12)

DEMANDE DE BREVET D'INVENTION

A1

(22) Date de dépôt : 28.07.94.

(30) Priorité :

(71) Demandeur(s) : FRANCE TELECOM
ETABLISSEMENT PUBLIC — FR et LA POSTE —
FR..

(43) Date de la mise à disposition du public de la
demande : 06.02.98 Bulletin 98/06.

(56) Liste des documents cités dans le rapport de
recherche préliminaire : Ce dernier n'a pas été
établi à la date de publication de la demande.

(60) Références à d'autres documents nationaux
apparentés :

(72) Inventeur(s) : GIRAULT MARC et STERN JACQUES.

(73) Titulaire(s) : .

(74) Mandataire : SOCIETE DE PROTECTION DES
INVENTIONS.

(54) PROCEDE D'AUTHENTIFICATION A NOMBRE REDUIT DE BITS TRANSMIS.

(57) Procédé d'authentification à nombre réduit de bits
transmis.

On réduit considérablement le nombre de bits de l'engagement envoyé par l'entité à authentifier mais on augmente de quelques unités seulement le nombre de bits de l'élément tiré par l'entité authentifiante.

Application aux procédés d'authentification dits à connaissance nulle.

FR 2 752 122 - A1



**PROCEDE D'AUTHENTIFICATION
A NOMBRE REDUIT DE BITS TRANSMIS**

Domaine technique

5 La présente invention a pour objet un procédé d'authentification à nombre réduit de bits transmis. Elle trouve une application dans la cryptographie dite à clé publique. Dans de tels procédés, une entité à authentifier possède une clé secrète et une clé publique associée. Une entité authentifiante a uniquement besoin de cette clé publique pour réaliser l'authentification.

10 15 L'invention concerne plus précisément encore le domaine des procédés d'authentification à connaissance nulle ("zero-knowledge"). Cela signifie que l'authentification se déroule suivant un protocole qui, de façon prouvée, et sous des hypothèses reconnues comme parfaitement raisonnables par la communauté scientifique, ne révèle rien sur la clé secrète de l'entité à authentifier.

20 25 L'invention trouve une application dans tous les systèmes nécessitant d'authentifier des entités ou des messages, ou de signer des messages, et plus particulièrement dans les systèmes où le nombre de bits transmis constitue un paramètre critique.

Etat de la technique antérieure

Dans tous les protocoles connus d'identification à connaissance nulle, l'entité à authentifier commence par fournir à l'entité authentifiante un ou plusieurs "engagements" (notés *c* ou éventuellement *x*) fonctions de paramètres choisis au hasard par l'entité à authentifier.

30 35 Dans un deuxième temps, l'entité authentifiante envoie à l'entité à authentifier un

paramètre ou "élément" noté e choisi au hasard (la "question"). Dans un troisième temps, l'entité à authentifier fournit à l'entité authentifierante un résultat correspondant y cohérent avec le ou les engagement(s) x ou c .

Dans ces applications, il est souvent important que le nombre total de bits transmis soit aussi petit que possible, afin notamment de réduire le temps de la communication et, dans certaines variantes 10 des protocoles de base (déescrites plus loin), de réduire le nombre de bits à stocker.

Le but de l'invention est de réduire de façon importante le nombre total de bits transmis dans 15 la plupart des protocoles d'identification connus, tout en conservant le même niveau de sécurité. Plus précisément, l'invention minimise le nombre de bits transmis dans le sens où, de façon prouvée, il est impossible de réduire encore ce nombre sans diminuer le 20 niveau de sécurité fourni par le protocole auquel on l'applique. Pour certains protocoles (par exemple celui de FIAT-SHAMIR (1) cette minimisation a un prix, à savoir l'augmentation du nombre de secrets que l'entité à authentifier doit détenir, ainsi qu'une augmentation 25 proportionnelle non négligeable du nombre de calculs à effectuer. Pour d'autres (notamment ceux de SCHNORR (3) et de GUILLOU-QUISQUATER (4)), cette minimisation n'agrave de façon sensible aucune autre caractéristique et s'avère donc particulièrement 30 intéressante.

A titre d'exemple, l'invention permet d'économiser environ 18% des bits transmis dans le protocole d'identification de SCHNORR (3), pour des choix classiques de longueurs de paramètres universels 35 et de niveaux de sécurité.

De façon générale, la plupart des protocoles d'identification à connaissance nulle existants, se déroulent en trois échanges, qui vont être décrits. On suppose, afin de simplifier la 5 description, que l'entité authentifiante B connaît déjà tous les paramètres publics caractéristiques de l'entité à authentifier A (identité, clé publique, etc.) et nécessaires à son identification.

Lors du premier échange, A fournit à B un 10 ou plusieurs engagements c . Lors du second échange, B envoie à A l'élément e . Lors du troisième échange, A fournit à B le résultat y correspondant à l'élément e , cohérent avec le ou les engagement(s) c . Dans certains protocoles, il y a deux échanges supplémentaires entre 15 A et B, consistant en un élément et un résultat de plus. C'est le cas du protocole de Shamir (2).

Le nombre u de questions possibles est directement relié au niveau de sécurité du protocole. Plus précisément, on peut montrer que, si le protocole 20 repose sur un problème mathématique difficile, et si les engagements sont de longueur suffisante (où la longueur désigne le nombre de bits), alors le niveau de sécurité, que l'on définit ici comme la probabilité de détection d'un imposteur (c'est-à-dire d'une entité C 25 qui tente frauduleusement de se faire passer pour A), est égal à $1-(1/u)$.

Souvent, on cherche à rendre le nombre de bits transmis au cours du premier échange aussi faible que possible (les engagements sont aussi courts que 30 possible), de façon à réduire le temps et le coût de l'authentification. Cependant, il peut être démontré que la longueur d'un engagement ne peut être choisie sous un certain seuil sans contrepartie, sauf à amoindrir le niveau de sécurité du protocole. Ce seuil 35 dépend de l'environnement et plus particulièrement du

nombre d'opérations élémentaires réputé "impossible en pratique" à exécuter en un temps et avec des moyens informatiques se situant "à l'échelle humaine". De façon traditionnelle, ce nombre est souvent pris égal à 5 2^{64} auquel cas le seuil évoqué ci-dessus est d'environ 128. Pour des raisons de commodité, c'est ce choix qui sera fait dans la description qui suit, mais cette description pourrait facilement être adaptée à d'autres choix.

10 Avec ce choix, la limite des 128 bits pour c ne peut être franchie sans nuire à la sécurité. En particulier, une longueur d'engagement de 64 bits seulement permet à l'imposteur de (au minimum) doubler la probabilité de réussite de son imposture, et même, 15 dans certains cas, de rendre cette probabilité égale à 1.

L'invention permet de franchir la limite de 128 bits pour les engagements (jusqu'à environ 70 bits), sans pour autant diminuer le niveau de sécurité. 20 D'après ce qui a été exposé plus haut, ceci est impossible si le reste du protocole est inchangé : il y a donc nécessairement une contrepartie. L'intérêt de l'invention est que cette contrepartie est mineure : il suffit d'augmenter légèrement le nombre u de questions 25 possibles, ce qui, en pratique, se traduit par une augmentation de trois ou quatre bits de la longueur de e.

Exposé de l'invention

30 L'invention consiste donc à diminuer de façon importante la longueur des engagements (d'environ 60 bits) et augmenter légèrement la longueur des questions (de 3 ou 4 bits), le protocole modifié suivant l'invention ayant exactement le même niveau de 35 sécurité que le protocole non modifié.

L'invention s'applique également aux schémas d'authentification de messages et éventuellement de signatures numériques de messages, sous certaines conditions explicitées par la suite.

5

De façon précise, l'invention a pour objet un procédé d'authentification à nombre réduit de bits transmis, entre une première entité dite à authentifier et une seconde entité dite authentifiante, ce procédé comprenant les opérations suivantes :

- 10 a) l'entité à authentifier choisit au hasard un nombre entier r , calcule un nombre appelé engagement x ou c fonction du nombre entier r et envoie cet engagement x ou c à l'entité authentifiante, cet engagement comprenant un certain nombre de bits,
- 15 b) l'entité authentifiante reçoit l'engagement x ou c , choisit au hasard un nombre e dans un certain intervalle, ce nombre étant appelé "élément" et ayant un certain nombre de bits et envoie cet élément e à l'entité à authentifier,
- 20 c) l'entité à authentifier reçoit l'élément e , effectue un calcul utilisant cet élément e et envoie le résultat y à l'entité authentifiante,
- 25 d) l'entité authentifiante reçoit le résultat y , effectue un calcul utilisant le résultat y et vérifie que le résultat de ce calcul est identique à l'engagement reçu x ou c auquel cas la première entité est authentifiée ;
- 30 un niveau de sécurité égal à $1 - \frac{1}{2^k}$ pouvant être défini pour cette authentification, à supposer que l'engagement x ou c possède un certain nombre minimum N de bits et l'élément e un certain nombre k de bits, ce procédé étant caractérisé par le fait que :

- le nombre de bits de l'engagement x ou c est pris très inférieur à N ;
- le nombre de bits de l'élément e est pris un peu supérieur à k ;
- 5 le niveau de sécurité restant alors le même.

Exposé détaillé de modes de réalisation

La définition précédente de l'invention s'applique notamment à trois protocoles connus, qui 10 vont maintenant être décrits à titre d'exemples non limitatifs, à savoir, respectivement, le protocole de SCHNORR, celui de GUILLOU-QUISQUATER et celui de FIAT-SHAMIR.

15 **1) PROTOCOLE DE SCHNORR**

Le protocole d'identification de C.P. SCHNORR (3) est basé sur la difficulté de calculer des logarithmes discrets. Les paramètres universels (c'est-à-dire ceux partagés par tous les utilisateurs) sont :

- 20 - un grand nombre premier p ,
- un nombre premier q tel que q soit un nombre premier divisant $p-1$ ou soit égal à $p-1$,
- un entier α (la "base") tel que $\alpha^q \equiv 1 \pmod{p}$,
- un petit entier k .

25

Les longueurs recommandées pour n et q sont respectivement (au moins) de 512 bits et 140 bits ; k détermine le nombre u de questions possibles par la relation $u=2^k$. Une valeur typique de k est 40 (ou 72 30 pour le schéma de signature correspondant).

La clé secrète d'un utilisateur est un entier s pris dans l'intervalle $\{1 \dots q\}$. Sa clé publique est $v=\alpha^{-s} \pmod{p}$. Le protocole est le suivant :

a) L'entité à authentifier choisit au hasard un entier r dans l'intervalle $\{1\dots q\}$, calcule $x=\alpha^r \pmod{p}$ et envoie x au vérificateur.

5 b) L'entité authentifiante choisit au hasard un élément e dans l'intervalle $\{0\dots 2^k-1\}$ et envoie e à l'entité à authentifier.

c) L'entité à authentifier calcule $y=r+se \pmod{q}$ et envoie y à l'entité authentifiante.

10 d) L'entité authentifiante contrôle que $x=\alpha^y v^e \pmod{p}$.

On remarque qu'un imposteur (qui ignore s) peut facilement tromper une entité authentifiante avec une probabilité égale à 2^{-k} , en choisissant un entier y , un élément e et en calculant x comme à l'étape d). Comme on peut prouver que, si le problème du logarithme discret est difficile, il ne peut实质iellement améliorer cette probabilité, le niveau de sécurité est donc égal à $1-2^{-k}$.

20 Afin de diminuer le nombre de bits transmis, l'auteur du protocole a suggéré d'envoyer à l'étape a) l'engagement $c=h(x)$ où h est une fonction pseudo-aléatoire. L'équation de vérification de l'étape d) devient alors :

25 $c=h(\alpha^y v^e \pmod{p})$.

Afin de conserver un niveau de sécurité égal à $1-2^{-k}$, il est nécessaire, toutes choses égales par ailleurs, que la longueur de c soit au moins de 128 bits.

30 Dans le protocole modifié selon l'invention, la longueur de l'engagement de c est réduite par exemple à 70 bits et cette réduction est compensée par une augmentation de la longueur de e de trois bits seulement. Une autre possibilité est de 35 choisir 69 bits pour c et quatre bits d'augmentation

pour e . Il peut alors être démontré que le niveau de sécurité du protocole ainsi modifié reste égal à $1-2^{-k}$. On a ainsi réduit le nombre total de bits transmis de 55 bits. Si l'on prend $k=40$, alors ce nombre total est 5 égal à : $70+(40+3)+160=273$ au lieu de $128+40+160=328$ avec des engagements de 128 bits, soit un gain d'environ 18%.

L'invention est particulièrement bien adaptée à ce protocole, en ce sens qu'elle n'entraîne 10 aucun changement des paramètres universels principaux, à savoir p , q , α , ni même des paramètres individuels (clés secrète et publique de l'utilisateur). De plus, elle est particulièrement utile dans un mode 15 d'utilisation particulier de ce protocole, qui consiste à stocker par avance les engagements. Ce mode d'utilisation est utilisé lorsque le dispositif de sécurité de l'entité à authentifier ne dispose pas des ressources lui permettant d'effectuer des opérations modulo un nombre de 512 bits (premier cas), ou bien les 20 effectue en un temps trop élevé pour que l'étape a) puisse être exécutée au moment même de l'authentification (deuxième cas).

Dans le premier cas, les engagements sont calculés par une autorité de confiance puis stockés 25 dans la carte : le dispositif ne peut alors être authentifié qu'un nombre limité de fois, égal au nombre d'engagements stockés. Cependant, lorsque ce nombre est atteint, il peut, par une procédure de recharge 30 d'engagements (éventuellement à distance), être à nouveau capable d'effectuer des authentifications. Dans le deuxième cas, les engagements sont calculés par le dispositif de l'entité à authentifier préalablement à l'authentification. Dans les deux cas, afin de ne pas avoir à stocker les nombres aléatoires r 35 correspondants, ces nombres sont avantageusement

produits par un générateur pseudo-aléatoire contenu dans le dispositif de sécurité de l'entité à authentifier.

A titre d'exemple, si le dispositif de sécurité de l'entité à authentifier est une carte à microprocesseur standard, alors l'étape a) ne peut être réalisée en un temps satisfaisant par la carte et même, dans certains cas, ne peut être réalisée du tout. Il est alors nécessaire de recourir au mode d'utilisation évoqué ci-dessus, et il est très avantageux d'utiliser l'invention, qui ne requiert que le stockage de 70 bits pour une authentification. Si l'on peut disposer de 1Koctet de mémoire reprogrammable à cette fin, alors la carte pourra effectuer 117 authentifications, après quoi il faudra recharger de nouvelles valeurs d'engagements.

Cette variante est encore plus intéressante dans le cas du protocole décrit dans la demande de brevet français n°94 01271 du 4 février 1994 intitulée "Procédé de signature numérique et d'authentification de messages utilisant un logarithme discret", car, dans ce protocole, l'étape c) ne contient pas d'opération modulo, ce qui fait que ce type d'opération n'a pas besoin d'être réalisée dans la carte à microprocesseur.

2) PROTOCOLE GUILLOU-QUISQUATER

Le protocole d'identification de GUILLOU et QUISQUATER (4) est basé sur la difficulté de factoriser des grands entiers. Dans la version dite "basée sur l'identité" de ce protocole, une autorité de confiance est utilisée pour calculer les clés secrètes des utilisateurs. Les paramètres universels sont :

- un grand entier non premier n (dont les facteurs premiers sont connus uniquement de l'autorité de confiance),
- un entier premier u de nombre de bits k .

5 La taille recommandée pour n est au minimum de 512 bits. Une valeur typique de k est 40 bits (ou 72 pour le schéma de signature correspondant).

10 La clé publique de l'entité à authentifier est donc "identité" (c'est-à-dire une chaîne binaire I contenant des informations à son propos et/ou à propos de son dispositif de sécurité). Sa clé secrète est la valeur s telle que $s^u I = 1 \pmod{n}$.

Le protocole de base est le suivant :

- a) L'entité à authentifier choisit au hasard un entier r dans l'intervalle $\{0..n\}$, calcule $x = r^u \pmod{n}$ et envoie x à l'entité authentifiante,
- b) l'entité authentifiante choisit au hasard un élément e dans l'intervalle $\{0, u-1\}$ et envoie e à l'entité à authentifier,
- c) l'entité à authentifier calcule $y = r s^e \pmod{n}$ et envoie y à l'entité authentifiante,
- d) l'entité authentifiante contrôle :

$$x = y^u I^e \pmod{n}.$$

25

On remarque qu'un imposteur (qui ignore s) peut facilement tromper une entité authentifiante avec une probabilité égale à $1/u$, en choisissant un entier y , un élément e et en calculant x comme dans l'étape d. 30 Comme on peut prouver que, si le problème de la factorisation est difficile, il ne peut实质iellement améliorer cette probabilité, le niveau de sécurité est donc égal à $1 - (1/u)$, qui est de l'ordre de $1 - 2^{-k}$.

Afin de diminuer le nombre de bits transmis, on peut envoyer à l'étape a) l'engagement $c=h(x)$ où h est une fonction pseudo-aléatoire. L'équation de vérification de l'étape d) devient 5 alors :

$$c=h(y^u I^e \pmod n).$$

Afin de conserver un niveau de sécurité du protocole de base égal à $1-2^{-k}$, il est nécessaire, toutes choses égales par ailleurs, que la longueur de c 10 soit au moins de 128 bits.

Dans le protocole modifié selon l'invention, la longueur de c est réduite par exemple à 70 bits et cette réduction est compensée par une augmentation de la longueur de e de trois bits 15 seulement, ce qui implique une augmentation d'autant de la longueur de u . Une autre possibilité est de choisir 69 bits pour c et quatre bits d'augmentation pour e . Il peut alors être démontré que le niveau de sécurité du protocole de base ainsi modifié reste égal à $1-2^{-k}$. On 20 a ainsi réduit le nombre total de bits transmis de 55 bits. Si l'on prend $k=40$, alors ce nombre total est égal à $70+(40+3)+512=625$ au lieu de $128+40+512=680$ avec des engagements de 128 bits, soit un gain d'environ 8%.

Les variantes décrites dans le cas du 25 protocole de SCHNORR sont encore applicables ici, mais sont moins intéressantes en ce sens que l'étape c) consiste en une opération modulaire importante qui, contrairement à celle de l'étape a), ne peut être effectuée à l'avance.

30

3°) PROTOCOLE FIAT-SHAMIR

Le protocole d'identification de FIAT et SHAMIR (1) est basé sur la difficulté de factoriser des grands entiers. Dans la version dite "basée sur 35 l'identité" de ce protocole, une autorité de confiance

est utilisée pour calculer les clés secrètes des utilisateurs. Les paramètres universels sont :

- un grand entier non premier n (dont les facteurs premiers sont connus uniquement de l'autorité de confiance),
- deux petits entiers k et t ,
- une fonction pseudo-aléatoire f .

La taille recommandée pour n est au minimum de 512 bits. Les valeurs de k et t sont étroitement reliées au niveau de sécurité du protocole (comme il sera décrit plus loin) ; le nombre u de questions possibles et le paramètre k sont reliés par la relation $u=2^k$. Des valeurs typiques de k et t sont 8 et 5 (ou 9 et 8 pour le schéma de signature correspondant).

La clé publique de l'entité à authentifier est son "identité" (c'est-à-dire une chaîne binaire I contenant des informations à son propos et/ou à propos de son dispositif de sécurité). Sa clé secrète est constituée de k valeurs s_j calculées comme suit : soit $v_j=f(I_j)$ pour k petites valeurs de j telles que v_j est un carré dans l'ensemble des entiers modulo n (pour la commodité de la description, on supposera que $j=1..k$). Alors $s_j^2v_j=1 \pmod{n}$ pour toute valeur de j .

Le protocole de base est le suivant :

a) l'entité à authentifier choisit au hasard un entier r dans l'intervalle $\{0..n\}$, calcule $x=r^2 \pmod{n}$ et envoie x à l'entité authentifiante,

b) l'entité authentifiante choisit au hasard un élément $e=(e_1, e_2, \dots, e_k)$ dans $\{0,1\}^k$ et envoie e à l'entité à authentifier,

c) l'entité à authentifier calcule $y=r \prod_{e_j=1} s_j \pmod{n}$

et envoie y à l'entité authentifiante,

d) l'entité authentifiante calcule tous les v_j et contrôle que : $x=y^2 \prod_{e_j=1} v_j \pmod{n}$.

On remarque qu'un imposteur (qui ignore s) peut facilement tromper une entité authentifiante avec une probabilité égale à 2^{-k} , en choisissant un entier y , un élément e et en calculant x comme dans l'étape d). Comme on peut prouver que, si le problème de la factorisation est difficile, il ne peut实质iellement améliorer cette probabilité, il suffit de répéter t fois le protocole de base pour obtenir un niveau de sécurité égal à $1-(1/u)^t=1-2^{-kt}$.

Afin de diminuer le nombre de bits transmis, les auteurs du protocole ont suggéré d'envoyer à l'étape a) l'engagement : $c=h(x)$ où h est une fonction pseudo-aléatoire. L'équation de vérification de l'étape d) devient alors :

$$c=h(y^2 \prod_{e_j=1} v_j \pmod{n})$$

Afin de conserver un niveau de sécurité du protocole de base égal à $1-2^{-k}$, il est nécessaire, toutes choses égales par ailleurs, que la longueur de c soit au moins de 128 bits.

Dans le protocole modifié selon l'invention, la longueur de c est réduite par exemple à 70 bits et cette réduction est compensée par une augmentation de la longueur de e de trois bits seulement. Une autre possibilité est de choisir 69 bits pour c et quatre bits d'augmentation pour e . Il peut alors être démontré que le niveau de sécurité du protocole de base ainsi modifié reste égal à $1-2^{-k}$. On a ainsi réduit le nombre total de bits transmis de 55 bits. Si l'on prend $k=8$ et $t=5$, alors ce nombre total est égal à : $70+(8+3)+512=593$ au lieu de $128+8+512=648$

avec des engagements de 128 bits, soit un gain d'environ 8,5%.

Cependant, un inconvénient de l'invention appliquée à ce protocole d'identification est qu'elle implique un plus grand nombre de secrets, puisque la longueur de e est précisément égale à ce nombre, à savoir k . De plus, le nombre de multiplications supplémentaires à effectuer par chacune des entités augmente proportionnellement de façon importante, et ce d'autant plus que k est choisi petit. Ainsi, l'invention comporte-t-elle dans ce cas des avantages et des inconvénients.

L'invention qui vient d'être décrite porte sur une authentification d'entité. Mais l'invention, s'applique à d'autres schémas comme l'authentification de messages et la signature numérique de messages.

Dans les schémas d'authentification de messages et de signature numérique de messages, l'entité à authentifier, en plus de prouver son identité, attache cette identité à un message donné. Cela se passe de manière interactive dans les schémas d'authentification de message et de manière non interactive dans les schémas de signature de message. L'invention est susceptible de s'appliquer surtout aux schémas d'authentification de messages dérivés de protocoles d'identification à connaissance nulle (en particulier les trois protocoles précités).

Le fait que l'invention s'applique ou non à ces schémas dépend de la façon exacte dont ils sont dérivés du protocole d'identification initial. La façon classique consiste à faire figurer le message M dans les paramètres y de la fonction pseudo-aléatoire h , ce qui donne :

35 $c=h(x, M)$ au lieu de $c=h(x)$.

Dans les schémas d'authentification de messages, le reste du protocole ne diffère pas, à l'adaptation évidente près de l'opération d) de vérification. Dans les schémas de signature, pour 5 lesquels l'entité signataire n'intervient pas dans les étapes a) et b), on choisit $e=c$, ce qui, a priori, enlève de l'intérêt à l'invention. Pour les deux types de schémas, il y a de toute façon la crainte que, si c est inférieur à 128 bits, alors l'entité à authentifier 10 ne soit capable, pour un nombre x issu du protocole, de trouver deux messages distincts M et M' aboutissant à la même valeur de c .

Si l'on se trouve dans un environnement où l'on n'a pas cette crainte (soit que la mise en oeuvre 15 du protocole dans le dispositif de sécurité rende cette attaque impossible, soit que l'on considère que les entités à authentifier sont dignes de confiance soit encore qu'une telle attaque serait sans intérêt pour eux), alors l'invention s'applique.

20 De façon générale, l'application de l'invention à d'autres schémas que des protocoles d'identification est possible, mais dépend des spécifications exactes de ces schémas ainsi que de leur réalisation pratique.

25 A titre d'exemple, le schéma d'authentification de message dérivé du protocole de SCHNORR est le suivant :

- a) l'entité à authentifier choisit au hasard un entier r dans $\{1..q\}$, calcule $x=\alpha^r \pmod p$, puis 30 $c=h(x, M)$ et envoie c à l'entité authentifiante,
- b) l'entité authentifiante choisit au hasard un élément e dans $\{0..2^{k-1}\}$ et l'envoie à l'entité à authentifier,
- c) l'entité à authentifier calcule $y=r+se \pmod q$ et 35 envoie y à l'entité authentifiante,

d) l'entité authentifiante contrôle que :
 $c=h(\alpha Y v^e \pmod{p}, M)$.

Dans un environnement où ce schéma peut
5 être modifié selon l'invention, alors le gain est
exactement le même que pour le protocole
d'identification modifié.

REFERENCES

1) A. FIAT et A. SHAMIR, "How to prove yourself : Practical solutions to identification and signature problems" Advances in Cryptology : Proceedings of CRYPTO'86, Lecture Notes in Computer Science, vol. 263, Springer-Verlag, Berlin, 1987, pp. 186-194.

5

10) 2) A. SHAMIR, "An efficient identification scheme based on permuted kernels" Advances in Cryptology : Proceedings of CRYPTO'89, Lecture Notes in Computer Science, vol. 435, Springer-Verlag, Berlin, 1987, pp. 606-609.

15

15) 3°) C.P. SCHNORR, "Efficient identification and signatures for smart cards" Advances in Cryptology : Proceedings of CRYPTO'89, Lecture Notes in Computer Science, vol. 435, Springer-Verlag, Berlin, 1987, pp. 239-252.

20

25) 4°) L.C. GUILLOU et J.J. QUISQUATER, "A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory" Advances in Cryptology : Proceedings of EUROCRYPT'88, Lecture Notes in Computer Science, vol. 330, Springer-Verlag, Berlin, 1988, pp. 123-128.

REVENDICATIONS

1. Procédé d'authentification à nombre réduit de bits transmis, entre une première entité dite à authentifier et une seconde entité dite authentifiante, ce procédé comprenant les opérations suivantes :

- a) l'entité à authentifier choisit au hasard un (des) paramètre(s) r , calcule un (des) nombre(s) appelé(s) engagement(s) c fonction(s) du (des) paramètre(s) r et envoie cet (ces) engagement(s) c à l'entité authentifiante, cet (ces) engagement(s) comprenant un certain nombre de bits,
- 10 b) l'entité authentifiante reçoit le (ou les) engagement(s) c , choisit au hasard un nombre e dans un certain ensemble, ce nombre étant appelé "élément" et ayant un certain nombre de bits et envoie cet élément e à l'entité à authentifier,
- 15 c) l'entité à authentifier reçoit l'élément e , effectue un (des) calcul(s) utilisant cet élément e et envoie le (les) résultat(s) y à l'entité authentifiante,
- 20 d) l'entité authentifiante reçoit le résultat y , effectue un calcul utilisant le (les) résultat(s) y et vérifie que ce calcul redonne le (les) engagement(s) reçu(s) c auquel cas la première entité est authentifiée ;
un niveau de sécurité égal à $1 - \frac{1}{u}$ pouvant être
- 25 30 défini pour cette authentification, à supposer que le (les) engagement(s) c possède(nt) un certain nombre minimum N de bits et que l'élément e fait partie d'un certain ensemble à u éléments et possède un certain nombre k de bits,

ce procédé étant caractérisé par le fait que :

- le nombre de bits de (des) engagement(s) c est pris très inférieur à N ;
- l'ensemble dans lequel est choisi l'élément e est pris un peu plus grand, ce qui se traduit par un nombre de bits de l'élément e un peu supérieur à k ;

le niveau de sécurité restant alors le même.

2. Procédé d'authentification selon la revendication 1, comprenant les opérations suivantes :

- a). l'entité à authentifier choisit au hasard un entier r compris entre 1 et un nombre q et calcule un nombre x égal à α^r (modulo p) où α est un entier appelé base tel que $\alpha^q = 1$ (modulo p), et où q est soit un nombre premier divisant $p-1$ soit égal à $p-1$ où p est un grand nombre premier ; l'entité à authentifier envoie une fonction $c=h(x)$ du nombre x où h est une fonction pseudo-aléatoire), le nombre envoyé c constituant l'engagement,
- b). l'entité authentifiante B choisit au hasard un élément e ayant un certain nombre de bits et envoie cet élément à l'entité à authentifier,
- c). l'entité à authentifier calcule un nombre y égal à $r + s.e$ (modulo q), où s est la clé secrète de l'entité à authentifier, s étant un entier compris entre 1 et q , et envoie le nombre y à l'entité authentifiante,
- d). l'entité authentifiante reçoit le nombre y , calcule $\alpha^y v^e$ (modulo p) où v est la clé publique de l'entité à authentifier, soit $v=\alpha^{-s}$ (modulo p) et vérifie $c=h(\alpha^y v^e \text{ modulo } p)$; un niveau de sécurité égal à $1 - \frac{1}{2^k}$ pouvant être défini en supposant pour l'engagement c un nombre minimum de bits égal à 128 et pour l'élément e un nombre de bits égal à k ,

le procédé étant caractérisé par le fait que le nombre de bits de l'engagement c est inférieur à 128 de plusieurs dizaines d'unités et le nombre de bits de l'élément e est supérieur de quelques unités au nombre k , le niveau de sécurité restant le même.

3. Procédé d'authentification selon la revendication 2, caractérisé par le fait que le nombre de bits de l'engagement c est voisin de 70 et le nombre de bits de l'élément e est voisin de 43, le niveau de sécurité obtenu étant le même qu'avec un engagement c de 128 bits et un élément e de 40 bits.

4. Procédé d'authentification selon la revendication 2, caractérisé par le fait que, dans l'opération c) l'entité à authentifier calcule un nombre y égal à $r + se$.

5. Procédé d'authentification selon la revendication 1 dans lequel :

a). l'entité à authentifier choisit au hasard un entier r compris entre 0 et n , où n est un grand entier non premier et calcule $x = r^u$ (modulo n) où u est un entier ayant une certaine longueur et envoie à l'entité authentifiante le nombre $c=h(x)$ où h est une fonction pseudo-aléatoire, le nombre envoyé c constituant l'engagement.

b). l'entité authentifiante choisit au hasard un élément e compris entre 0 et $u-1$ et envoie e à l'entité à authentifier,

c). l'entité à authentifier calcule $y=rs^e$ (modulo n) où s est la clé secrète de l'entité à authentifier telle que $s^{uI}=1$ (modulo n), où I est la clé publique de l'entité à authentifier, et l'entité à authentifier envoie le nombre y à l'entité authentifiante,

d). l'entité authentifiante reçoit le nombre y , calcule $h(y^{UI} \bmod n)$ et vérifie que l'on retrouve l'engagement,

un niveau de sécurité égal à $1 - \frac{1}{2^k}$ pouvant être défini

5 en supposant pour l'engagement c un nombre minimum de bits (N) égal à 128 et en supposant pour l'élément e un nombre de bits égal à k ,

ce procédé étant caractérisé par le fait que le nombre de bits de l'engagement c est inférieur à 128 de 10 plusieurs dizaines d'unités, et le nombre de bits de l'élément e est supérieur de quelques unités au nombre k , le niveau de sécurité restant le même.

6. Procédé d'authentification selon la revendication 5, caractérisé par le fait que le nombre 15 de bits de l'engagement c est voisin de 70 et le nombre de bits de l'élément e voisin de 43, le niveau de sécurité obtenu étant le même qu'avec un engagement de 128 bits et un élément e de 40 bits.

7. Procédé d'authentification selon la 20 revendication 1, comprenant les opérations suivantes :

a). l'entité à authentifier choisit au hasard un entier r compris entre 0 et n où n est un grand entier non premier, calcule $x=r^2 \pmod n$ et envoie à l'entité authentifiante le nombre $c=h(x)$ 25 où h est une fonction pseudo-aléatoire, le nombre envoyé c constituant l'engagement,

b). l'entité authentifiante choisit au hasard un élément e comprenant un certain nombre de bits et envoie e à l'entité à authentifier,

30 c). l'entité à authentifier calcule le produit par r d'un produit de valeurs s_j où l'ensemble des s_j pour toute valeur de j constitue la clé secrète de l'entité à authentifier, et l'entité à authentifier envoie le résultat y de ce calcul à l'entité authentifiante,

d). l'entité authentifiante calcule la valeur du produit par y^2 du produit des v_j (modulo n), applique au résultat ainsi trouvé la fonction h et vérifie que le résultat trouvé s'identifie à l'engagement c ,

5 un niveau de sécurité égal à $1 - \frac{1}{2^k}$ pouvant être défini à supposer, pour l'engagement c , un nombre minimum de bits égal à 128 et pour l'élément e un nombre de bits égal à k , en répétant t fois les opérations a) à d),
10 ce procédé étant caractérisé par le fait que le nombre de bits de l'engagement c est inférieur à 128 de plusieurs dizaines de bits et le nombre de bits de l'élément e est supérieur de quelques unités au nombre k , le niveau de sécurité restant le même.

15 8. Procédé d'authentification selon la revendication 7, caractérisé par le fait que le nombre de bits de l'engagement c est voisin de 70, le nombre k de bits de l'élément e voisin de 11 et le nombre t voisin de 5.

20 9. Procédé d'authentification selon l'une quelconque des revendications 1 à 7, caractérisé par le fait que l'entité authentifiante effectue en outre une authentification d'un message M émis par l'entité à authentifier, l'entité à authentifier calculant dans 25 l'opération a un engagement c égal à $h(x, M)$ où M est le message à authentifier, et l'entité authentifiante vérifiant, dans l'opération d) que la fonction h du calcul qu'elle effectue et du message M qu'elle a reçu est identique à l'engagement c .

30 10. Procédé selon la revendication 1, caractérisé par le fait qu'il comprend des échanges supplémentaires entre la première et la seconde entités consistant chacun en l'échange d'un élément e supplémentaire et d'un résultat y supplémentaire.